"Express Mail" mailing label number **EV 314841675 US**

Date of Deposit **October 13, 2003**

Case No. 9683/160

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:		)	
	Dai KAMIYA	)	
	Kazuhiro YAMADA	)	
	Takashi KONDO	)	Examiner: To Be Assigned
Serial No.:	To Be Assigned	)	Group Art Unit No.: To Be Assigned
Filing Date:	October 13, 2003	)	

For: COMMUNICATION DEVICE, PROGRAM AND RECORDING MEDIA

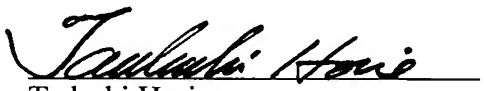
SUBMISSION OF CERTIFIED COPY OF FOREIGN PRIORITY DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicants submit herewith a certified copy of Japanese Patent Application No. JP 2002-316635 filed October 30, 2002, to which the above-identified United States Patent Application claims the right of foreign priority under 35 U.S.C. § 119.

Respectfully submitted,



Tadashi Horie
Registration No. 40,437
Agent for Applicants

BRINKS HOFER GILSON & LIONE
P.O. BOX 10395
CHICAGO, ILLINOIS 60610
(312) 321-4200

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２００２年１０月３０日

出 願 番 号
Application Number: 特願２００２－３１６６３５

[ST. 10/C]: ［ＪＰ２００２－３１６６３５］

出 願 人
Applicant(s): 株式会社エヌ・ティ・ティ・ドコモ

２００３年 ９月２２日

特許庁長官
Commissioner,
Japan Patent Office 今井康夫

出証番号 出証特２００３－３０７７８９９

【書類名】　特許願
【整理番号】　DCMH140447
【提出日】　平成14年10月30日
【あて先】　特許庁長官　殿
【国際特許分類】　G06F　9/00
　　H04Q　7/28
【発明の名称】　通信装置、プログラムおよび記録媒体
【請求項の数】　12
【発明者】
　【住所又は居所】　東京都千代田区永田町二丁目１１番１号　株式会社エヌ・ティ・ティ・ドコモ内
　【氏名】　神谷　大
【発明者】
　【住所又は居所】　東京都千代田区永田町二丁目１１番１号　株式会社エヌ・ティ・ティ・ドコモ内
　【氏名】　山田　和宏
【発明者】
　【住所又は居所】　東京都千代田区永田町二丁目１１番１号　株式会社エヌ・ティ・ティ・ドコモ内
　【氏名】　近藤　隆
【特許出願人】
　【識別番号】　392026693
　【氏名又は名称】　株式会社エヌ・ティ・ティ・ドコモ
【代理人】
　【識別番号】　100098084
　【弁理士】
　【氏名又は名称】　川▲崎▼　研二

【選任した代理人】

【識別番号】　100111763

【弁理士】

【氏名又は名称】　松本　隆

【手数料の表示】

【予納台帳番号】　038265

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　通信装置、プログラムおよび記録媒体

【特許請求の範囲】

【請求項１】　データを記憶する記憶手段と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、

前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記識別情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、

前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、

前記特定手段に特定されたデータの使用について、前記オブジェクト生成手段により生成されたオブジェクトの使用のみを前記実行手段に許可するアクセス制御手段と

を有することを特徴とする通信装置。

【請求項２】　前記選択手段は、前記特定手段に特定されたデータを内包するオブジェクトの型を、前記取得手段により取得された前記識別情報によって示される確率が予め定められた基準確率より低い場合には前記完全カプセル化型とし、前記基準確率以上の場合には前記非完全カプセル化型とする

ことを特徴とする請求項１に記載の通信装置。

【請求項３】　データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶手段と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、

該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、

前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、

前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、

前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行手段に許可する使用オブジェクト制御手段と

を有することを特徴とする通信装置。

【請求項４】　前記使用オブジェクト制御手段は、前記取得手段により取得された前記識別情報によって示される確率が、予め定められた第１の基準確率より高い場合には、前記完全カプセル化型及び前記非完全カプセル化型のオブジェクトの使用を前記実行手段に許可し、

前記確率が、前記第１の基準確率以下であり、かつ予め定められた第２の基準確率より高い場合には、前記完全カプセル型のオブジェクトの使用を前記実行手段に許可し、

前記確率が、前記第２の基準確率以下である場合には、いずれのオブジェクトの使用も前記実行手段に許可しないことを特徴とする請求項３に記載の通信装置。

【請求項５】　データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶手段と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取

得する取得手段と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、

前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報及び前記識別情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、

前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、

前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行手段に許可する使用オブジェクト制御手段と

を有することを特徴とする通信装置。

【請求項６】　前記オブジェクト生成手段により生成された前記オブジェクトを前記プログラムの実行終了に応じて削除する削除手段をさらに有する

ことを特徴とする請求項１、３または５に記載の通信装置。

【請求項７】　パケット通信機能を有する携帯電話機であることを特徴とする請求項１、３または５に記載の通信装置。

【請求項８】　前記プログラムは、Javaプログラミング言語で記述されたプログラムであって、

前記プログラムを実行するためのJava実行環境を有していることを特徴とする請求項１、３または５に記載の通信装置。

【請求項９】　コンピュータに、

データを記憶する記憶機能と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、

前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記識別情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、

前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、

前記特定機能に特定されたデータの使用について、前記オブジェクト生成機能により生成されたオブジェクトの使用のみを前記実行機能に許可するアクセス制御機能と

を実現させるためのプログラム。

【請求項１０】　コンピュータに、

データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶機能と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、

前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、

前記特定機能に特定されたデータを内包する、前記選択機能によって選択され

た型のオブジェクトを生成するオブジェクト生成機能と、

前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行機能に許可する使用オブジェクト制御機能と

を実現させるためのプログラム。

【請求項１１】　コンピュータに、

データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶機能と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、

前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報及び前記識別情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、

前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、

前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行機能に許可する使用オブジェクト制御機能と

を実現させるためのプログラム。

【請求項１２】　請求項９乃至１１に記載のプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明は、通信装置に格納されたデータに対するセキュリティを確保するための技術に関する。

【０００２】

【従来の技術】

近年、パケット通信機能を有する携帯電話機を用いて、インターネットに接続されているサーバから様々なプログラムをダウンロードすることが一般化しつつある。

【０００３】

ところで、世界中の様々な人々は、インターネットにおいて、自由に情報の公開やプログラムの提供を行うことができる。インターネットは、このような利点を有する反面、欠点もある。例えば、悪意の有る個人や団体によって、通信装置内に記憶されているデータを密かに盗み出すプログラムが、インターネットを介して提供されてしまう場合がある。あるいは、悪意は無いものの通信装置において動作させると不具合を引き起こしてしまうプログラムが、インターネットを介して提供されてしまうことがある。

したがって、インターネットを介して提供されたプログラムに対して、通信装置の内部および外部のリソースを何ら制限することなくアクセスできるようにしてしまうと、例えば、通信装置内に記憶されているユーザの電話番号やメールアドレス、銀行口座番号などが勝手に読み出され、通信装置の外へ流出してしまうといった事態が生じ得る。

【０００４】

このため、例えば、Java（登録商標）言語で記述されたプログラムを実行することが可能な携帯電話機は、インターネットを介してダウンロードされたJavaプログラムを実行するときにアクセスできるリソースを、このプログラムのダウンロード元のサーバと、このプログラムに対して割り当てられた記憶領域のみに制限し、それ以外のユーザの電話番号やメールアドレス、電話帳データ等の個人情報等のリソースには一切アクセスすることができないようにしていた。一方、携帯電話機は、携帯電話機に記憶されている個人情報を、ネイティブプログラムのみで操作するようにして、個人情報のセキュリティを確保していた（例えば、非

特許文献１参照）。なお、ここでは、携帯電話機の販売以前にそのメモリに書き込まれるプログラムをネイティブプログラムと呼ぶ。

【０００５】

【非特許文献１】

ｉアプリコンテンツ開発ガイド　ｆｏｒ　５０４ｉ　詳細編　インターネット＜ＵＲＬ：http://www.nttdocomo.co.jp/p_s/imode/java/＞

【０００６】

【発明が解決しようとする課題】

しかしながら、上述したアクセス制限の仕組みは、携帯電話機におけるセキュリティを確保する上で一定の効果を奏するものの、ダウンロードされたプログラムに対して様々な動作制限を課すことになり、プログラムの多様化を阻む要因の一つになっていた。

本発明は、以上説明した事情に鑑みてなされたものであり、携帯電話機等の通信装置に記憶されているデータのセキュリティを確保しつつ、当該データを使用する様々なプログラムを提供できるようにする技術を提供することを目的としている。

【０００７】

【課題を解決するための手段】

上記課題を解決するために、この発明は、データを記憶する記憶手段と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記識別情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを

生成するオブジェクト生成手段と、前記特定手段に特定されたデータの使用について、前記オブジェクト生成手段により生成されたオブジェクトの使用のみを前記実行手段に許可するアクセス制御手段とを有することを特徴とする通信装置を提供する。

【０００８】

また、この発明は、データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶手段と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行手段に許可する使用オブジェクト制御手段とを有することを特徴とする通信装置を提供する。

【０００９】

また、本発明は、データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶手段と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報及び

前記識別情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行手段に許可する使用オブジェクト制御手段とを有することを特徴とする通信装置を提供する。

【００１０】

また、本発明は、コンピュータに、データを記憶する記憶機能と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記識別情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、前記特定機能に特定されたデータの使用について、前記オブジェクト生成機能により生成されたオブジェクトの使用のみを前記実行機能に許可するアクセス制御機能とを実現させるためのプログラムを提供する。

【００１１】

また、本発明は、コンピュータに、データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶機能と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、内包したデータを呼び出し元へ

引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行機能に許可する使用オブジェクト制御機能とを実現させるためのプログラムを提供する。

【００１２】

また、本発明は、コンピュータに、データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶機能と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報及び前記識別情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行機能に許可する使用オブジェクト制御機能とを実現させるためのプログラムを提供する。

【００１３】

この発明によれば、通信装置は、プログラムと該プログラムの識別情報を受信し、前記プログラムを実行した場合に使用されるデータを特定し、前記識別情報に基づいて生成するオブジェクトの型（非完全カプセル化型、または完全カプセル化型）を選択して、選択された型のオブジェクトを生成し、前記プログラムを

実行するときには、前記生成されたオブジェクトのみを使用して前記データを使用する。

【００１４】

【発明の実施の形態】

［１．第１実施形態］

以下、図面を参照して本発明の第１実施形態について説明する。なお、各図において共通する部分には、同一の符号が付されている。

【００１５】

［１－１．実施形態の構成］

＜１－１－１．通信システムの構成＞

図１は、この発明の実施形態に係る通信システム１の構成を示すブロック図である。同図に示すように通信システム１は、コンテンツサーバ１０と、インターネット２０と、移動パケット通信網３０と、携帯電話機４０とを有している。なお、この通信システム１には、本来、多数の携帯電話機４０が収容されるが、図面が煩雑になることを防ぐため、図１には、１つの携帯電話機４０のみを図示している。また、同様の理由により、図１には、それぞれ１つのコンテンツサーバ１０、ゲートウェイサーバ３１および基地局３２のみを図示している。

【００１６】

コンテンツサーバ１０は、インターネット２０および移動パケット通信網３０を介して携帯電話機４０とパケット通信を行う機能を有している。このコンテンツサーバ１０には、携帯電話機４０に提供するプログラムや画像データ、楽曲データなどの種々のコンテンツが格納されている。これらのコンテンツの中には、携帯電話機４０において実行可能なJavaアプリケーションプログラム（以下、JavaＡＰと略称する）がある。

【００１７】

移動パケット通信網３０は、当該移動パケット通信網３０に収容される携帯電話機４０に対してパケット通信サービスを提供する通信網である。ゲートウェイサーバ３１は、移動パケット通信網３０とインターネット２０とのデータの授受を中継する。また、基地局３２は、移動パケット通信網３０の通信サービスエリ

ア内に多数設置されており、携帯電話機４０と無線通信を行う。

【００１８】

携帯電話機４０は、基地局３２と無線通信を行う。また、この携帯電話機４０は、移動パケット通信網３０およびインターネット２０を介してコンテンツサーバ１０とパケット通信を行う機能を有しており、コンテンツサーバ１０からコンテンツをダウンロードすることができる。

【００１９】

＜１－１－２．携帯電話機の構成＞

図２は、携帯電話機４０のハードウェア構成を示すブロック図である。同図に示すように携帯電話機４０は、無線通信部４０１と、操作入力部４０２と、通話処理部４０３と、通信インタフェース４０４と、ＣＰＵ４０５と、液晶表示部４０６と、記憶部４０７とを有しており、これらの各部はバス４１１により接続されている。

【００２０】

無線通信部４０１は、アンテナ４０１ａを備え、基地局３２との間で行われる無線通信を制御する。この無線通信部４０１は、ＣＰＵ４０５の制御の下、送話音声に関するデータやパケット通信用のデータなどを搬送波に重畳して送信信号を生成し、この信号を基地局３２へ送信する。また、無線通信部４０１は、基地局３２から送られてくる無線信号をアンテナ４０１ａを介して受信し、この信号を復調して自機４０宛の受話音声に関するデータやパケット通信用のデータなどを得る。

【００２１】

操作入力部４０２は、数字や文字、操作指示などを入力するための複数のキーを有しており、これらのキーの操作に応じた操作信号をＣＰＵ４０５に出力する。また、通話処理部４０３は、例えば、マイクロフォンやスピーカ、音声処理部などを有しており、ＣＰＵ４０５の制御の下、呼の接続／切断を含む通話処理を行う。

【００２２】

通信インタフェース４０４は、通信ケーブルを介して接続された電子機器との

有線通信を制御する。また、ＣＰＵ４０５は、記憶部４０７に格納されている各種プログラムを実行することにより、バス４１１を介して接続されている装置各部を制御する。また、液晶表示部４０６は、液晶表示パネルと、この液晶表示パネルの表示制御を行う駆動回路とを有している。

【００２３】

記憶部４０７は、ＲＯＭ４０８と、ＲＡＭ４０９と、例えば、ＳＲＡＭ（Static－RAM）やＥＥＰＲＯＭ（Electrically Erasable Programmable－ROM）などの不揮発性メモリ４１０とを有している。ＲＯＭ４０８には、例えば、携帯電話機４０用のオペレーティングシステム（以下、ＯＳと略称する）やＷｅｂ（World Wide Web）ブラウザ等のソフトウエア、Java実行環境を構築するためのソフトウェアが記憶されている。また、ＲＡＭ４０９は、ＣＰＵ４０５のワークエリアとして用いられ、ＣＰＵ４０５により実行される各種のプログラムやデータが一時的に記憶される。

【００２４】

不揮発性メモリ４１０には、携帯電話機４０の製品出荷時点から当該携帯電話機４０用のプログラムが記憶される。また、不揮発性メモリ４１０には、コンテンツサーバ１０からダウンロードされたJavaＡＰなどのコンテンツが記憶される。加えて、この不揮発性メモリ４１０には、電話番号を表すデータやメールアドレスを表すデータなどのデータを含んでいるアドレス帳データ、受信あるいは送信した電子メールデータ、着信や発信に関する履歴データ、電子決済を行うためのユーザの銀行口座番号を表すデータやクレジットカード番号を表すデータなどの各種データが記憶される。

【００２５】

なお、以下、本明細書では、携帯電話機４０の製品出荷時点において既にＲＯＭ４０８や不揮発性メモリ４１０に記憶されているプログラムを、ダウンロードされたJavaＡＰと区別するため、ネイティブプログラムと記載する。このネイティブプログラムには、自身がネイティブプログラムであることを示す識別情報が付与されている。

【００２６】

また、不揮発性メモリ４１０は、ＪＡＲストレージ４１０ａと、個別スクラッチパッド４１０ｂと、共通スクラッチパッド４１０ｃとを有している。

【００２７】

ここで、ＪＡＲストレージ４１０ａ、個別スクラッチパッド４１０ｂおよび共通スクラッチパッド４１０ｃについて説明する前に、まず、携帯電話機４０にダウンロードされるJavaＡＰについて説明する。JavaＡＰは、JavaＡＰの本体プログラムおよび当該本体プログラムの実行に応じて利用される画像ファイルや音声ファイルなどを１つにまとめたＪＡＲ（Java Archive）ファイルと、このＪＡＲファイルのインストールや起動、ネットワークアクセスなどを制御するための各種制御情報が記述されたＡＤＦ（Application Descriptor File）とを有している。ダウンロードされたＪＡＲファイルおよびＡＤＦは、不揮発性メモリ４１０に記憶される。本実施形態においては、ＡＤＦには、図３に示すように、JavaＡＰの名称を示す「AppName」や、インターネット２０におけるＪＡＲファイルのＵＲＬを示す「PackageURL」や、ＪＡＲファイルのサイズをしめす「AppSize」や、ＪＡＲファイルの最終更新日を示す「LastModified」等の従来からＡＤＦに内包されているデータに加えて、「トラステッドアプリケーション識別子」が内包されている。この「トラステッドアプリケーション識別子」とは、移動パケット通信網３０を運営する通信事業者やＣＡ（Certificate Authority）のような公正な第３者機関によりJavaＡＰの内容が審査され、一定の基準を満たしていると認定されたJavaＡＰとそれ以外のプログラムとを識別するためのデータである。一定の基準とは、例えば、プログラムが携帯電話機４０に記憶されているデータを外部に漏洩させることなく適正に取り扱い、携帯電話機４０において正常に動作する、等の基準である。上記の第３者機関は、移動パケット通信網３０を用いて提供される通信サービスに加入した全ての者に信頼されているから、この第３者機関により認定されたプログラムには、一定の信頼が化体する。よって、ここでは、このような一定の信頼が化体したプログラムを「トラステッドアプリケーション」とよび、それ以外のプログラムを「非トラステッドアプリケーション」とよぶ。「トラステッドアプリケーション識別子」は、その値が"0"である場合には、当該ＡＤＦファイルに対応するJavaＡＰが非トラステッドアプリケーシ

ョンであることを示し、”１”である場合には、当該ＡＤＦファイルに対応するJavaＡＰがトラステッドアプリケーションであることを示している。

【００２８】

ＪＡＲストレージ４１０ａおよび個別スクラッチパッド４１０ｂには、ダウンロードされたJavaＡＰ毎に当該JavaＡＰ用の記憶領域が設けられる。ＪＡＲストレージ４１０ａ内の各記憶領域には、JavaＡＰのＪＡＲファイルが記憶される。また、個別スクラッチパッド４１０ｂ内の各記憶領域には、例えば、JavaＡＰがゲームプログラムである場合、過去の得点データやセーブデータなど、JavaＡＰの利用に応じて発生した当該JavaＡＰ用のデータが記憶される。さらに、共通スクラッチパッド４１０ｃには、複数のJavaＡＰが共通して使用するデータが記憶される。

【００２９】

また、ダウンロードの後、JavaＡＰが携帯電話機４０において実行される場合、このJavaＡＰの実行に伴って携帯電話機４０がアクセスすることのできるリソースは、このJavaＡＰのダウンロード元のコンテンツサーバ１０と、このJavaＡＰに対して割り当てられたＪＡＲストレージ４１０ａおよび個別スクラッチパッド４１０ｂ内の記憶領域と、共通スクラッチパッド４１０ｃと、のみに制限され、それ以外のリソースにアクセスすることはできない。

【００３０】

＜１－１－３．Java実行環境＞

図４は、携帯電話機４０におけるJavaＡＰの実行環境を説明するための図である。同図に示すように本実施形態に係る携帯電話機４０には、JavaＡＰの実行環境を構築するためのソフトウェアとして、ＫＶＭ（K Virtual Machine）と、コンフィギュレーションとしてＣＬＤＣ（Connected Limited Device Configuration）と、プロファイルとして通信事業者が独自に策定したオリジナル拡張ライブラリとが記憶されている。

【００３１】

ＫＶＭは、小型電子機器用に設計変更されたＪＶＭ（Java Virtual Machine）であって、JavaＡＰの実行ファイル形式であるバイトコードをＣＰＵ４０５がＯ

Ｓを介して解釈／実行可能な命令コードに変換する。また、ＣＬＤＣクラスライブラリは、ＣＬＤＣ用のクラスライブラリである。

【００３２】

オリジナル拡張ライブラリは、ＣＬＤＣを基礎として携帯電話機に特化した機能を提供するためのクラスライブラリである。このオリジナルJava拡張ライブラリには、例えば、ユーザインタフェースＡＰＩ（Application Program Interface）、ネットワーキングＡＰＩ、スクラッチパッドＡＰＩ、完全カプセル化ＡＰＩ、非完全カプセル化ＡＰＩなどが含まれている。

【００３３】

ここで、ユーザインタフェースＡＰＩは、携帯電話機４０のユーザインタフェース機能をサポートするＡＰＩであり、ネットワーキングＡＰＩは、ＵＲＬ（Uniform Resource Locator）により指定されたネットワークリソースへのアクセスをサポートするＡＰＩである。また、スクラッチパッドＡＰＩは、個別スクラッチパッド４１０ｃや共通スクラッチパッド４１０ｄに対するデータの書き込みや読み出しをサポートするＡＰＩである。さらに、完全カプセル化ＡＰＩは、完全カプセル化オブジェクトを生成するためのＡＰＩであり、非完全カプセル化ＡＰＩは、非完全カプセル化オブジェクトを生成するためのＡＰＩである。

【００３４】

また、携帯電話機４０は、ＣＬＤＣクラスライブラリおよびオリジナル拡張ライブラリに加え、メーカ独自拡張ライブラリを有している。このメーカ独自拡張ライブラリは、携帯電話機４０を製造する各メーカがそれぞれ独自の機能を提供するためのクラスライブラリである。

【００３５】

次に、ＪＡＭ（Java Application Manager）は、ＯＳによる制御の下で、携帯電話機４０にダウンロードされたJavaＡＰや、完全カプセル化オブジェクト、非完全カプセル化オブジェクトなどを管理する機能を有している。例えば、ＪＡＭは、JavaＡＰのインストールや更新、削除を行う機能、不揮発性メモリ４１０に記憶されているJavaＡＰをリスト表示する機能、JavaＡＰの実行管理（起動や強制終了など）を行う機能、JavaＡＰの実行に伴う携帯電話機４０のアクセスを制

限する機能、完全カプセル化オブジェクトや非完全カプセル化オブジェクトの生成、更新、削除を行なう機能などを有している。

【００３６】

また、同図に示すように、電話帳機能やブラウザ機能、ネットワーク通信機能などを提供するネイティブプログラムは、ＯＳによる制御の下で直接動作する。

【００３７】

＜１－１－４．オブジェクトの構成＞

次に、オブジェクトについて説明する。オブジェクトとは、データ（Javaプログラミング言語においてはフィールド）と操作（Javaプログラミング言語においてはメソッド）が一体となったものである。Javaプログラミング言語では、「private」というアクセス修飾子を用いてオブジェクト内のフィールドをprivateフィールドに宣言することで、当該privateフィールドに記憶されるデータのカプセル化を図る。このカプセル化により、カプセル化オブジェクトが生成される。

図５は、カプセル化オブジェクトについて説明するための模式図である。同図に示すように、カプセル化オブジェクトとは、カプセル化された１以上のデータと、当該カプセル化された各データに対するオブジェクト外部からの操作を可能とするための１以上のメソッドとを有するオブジェクトである。

【００３８】

同図に示す例では、２つのデータ１，２と、２つのメソッド１，２とを有するカプセル化オブジェクトが例示されている。このカプセル化オブジェクト内のデータ１，２は共にカプセル化されているため、オブジェクトの外部からデータ１，２を直接読み書きすることはできない。したがって、ダウンロードされたプログラムがカプセル化オブジェクト内のデータ１，２に対してアクセスする場合、プログラムは、メソッド１，２を使用して目的のデータ１またはデータ２に対する操作をカプセル化オブジェクトに指令しなければならない。

【００３９】

ここで、同図に示すメソッド１が、例えば、指定されたデータ自体を操作元のプログラムへ引き渡すメソッドであれば、操作元のプログラムは、メソッド１を使用してカプセル化オブジェクト内の任意のデータ１，２を取得することが可能

である。また、同図に示すメソッド２が、例えば、指定されたデータを液晶画面に表示させるメソッドであれば、操作元のプログラムは、メソッド２を使用してカプセル化オブジェクト内の任意のデータ１，２を画面表示させることが可能である。ここで注目すべき点は、メソッド２を使用してカプセル化オブジェクト内の任意のデータ１，２を画面表示させたプログラムは、表示させるデータやメソッドを指定してカプセル化オブジェクトへ指令を出すものの、表示させるデータ自体を取得していない点である。

【００４０】

つまり、データそのものを操作元のプログラムに引き渡すメソッドを１つも有していないカプセル化オブジェクト（完全カプセル化オブジェクト）であれば、操作元のプログラムは、オブジェクト内のデータそのものを取得することはできないが、このオブジェクトに備わるメソッドを使用してオブジェクト内のデータ操作を行うことはできる。

【００４１】

したがって、操作元のプログラムが、非トラステッドアプリケーションである場合には、プログラムがアクセスするデータ全てを完全カプセル化オブジェクトとして扱うようにすれば、当該プログラムにデータ自体を引き渡すことがないので、携帯電話機４０に記憶されているデータのセキュリティを確保することができる。また、このように扱うことで、非トラステッドアプリケーションであったとしても、アドレス帳データや電子メールデータなど、従来はセキュリティを確保する観点から一切のアクセスを認めていなかったデータに対し、完全カプセル化オブジェクトが有するメソッドを用いて操作（アクセス）を行うことができる。

【００４２】

図６は、電話帳データに関する非完全カプセル化オブジェクトについて例示する模式図である。

上述したように、Javaプログラミング言語では、「private」というアクセス修飾子を用いてオブジェクト内のフィールドをprivateフィールドに宣言することで、当該privateフィールドに記憶されるデータのカプセル化を図る。つまり

、オブジェクト内のフィールドが全てprivateフィールドである場合、各privateフィールドに記憶されているデータをオブジェクトの外部から直接読み書きすることができなくなる。このようにした場合、各privateフィールドに記憶されているデータに対してオブジェクトの外部からアクセスするには、このオブジェクトに備わるメソッドを使用してデータに対する操作を当該オブジェクトに指令しなければならない。

【００４３】

同図に示す非完全カプセル化オブジェクトには、２つのprivateフィールドが設けられ、それぞれprivate char value[１]、private char value[２]という電話帳の文字列データが記憶されている。また、この非完全カプセル化オブジェクトは、getBytes()、drawString()という２つのメソッドを有している。ここで、getBytes()は、オブジェクト内のデータをバイト配列の形式で操作元のプログラムへ引き渡すメソッドである。したがって、ダウンロードされたJavaＡＰは、このgetBytes()というメソッドを使用して、非完全カプセル化オブジェクト内の電話帳の文字列データ（private char value[１]、private char value[２]）を取得することが可能である。加えて、JavaＡＰは、取得した電話帳の文字列データを当該JavaＡＰのダウンロード元のサーバ（コンテンツサーバ１０）へ送信することなどができる。

【００４４】

また、drawString()は、オブジェクト内のデータを携帯電話機４０の液晶画面に表示させるメソッドである。JavaＡＰは、このdrawString()というメソッドを使用して、非完全カプセル化オブジェクト内の電話帳の文字列データ（private char value[１]、private char value[２]）を液晶画面に表示させることもできる。

【００４５】

一方、図７は、電話帳データに関する完全カプセル化オブジェクトについて例示する模式図である。同図に示す完全カプセル化オブジェクトが図６に示した非完全カプセル化オブジェクトと異なるのは、完全カプセル化オブジェクトは、上述したgetBytes()のように、オブジェクト内のデータそのものを操作元のプログ

ラムへ引き渡すメソッドを有していない点である。

【００４６】

すなわち、完全カプセル化オブジェクトは、カプセル化された上に、オブジェクト内のデータそのものを操作元のプログラムへ引き渡すメソッドを１つも有していない。したがって、ダウンロードされたJavaＡＰは、drawString()というメソッドを使用してオブジェクト内の電話帳の文字列データ（private char value[１]、private char value[２]）を画面表示させることはできるが、電話帳の文字列データそのものを取得することはできない。以上のようなことから、非トラステッドアプリケーションが携帯電話機４０にダウンロードされた場合であっても、このような非トラステッドアプリケーションに電話帳データが引き渡されることはなく、当然、電話帳データがサーバなど携帯電話機４０の外部へ送信されることもない。

【００４７】

ところで、drawString()というメソッドを使用してオブジェクト内の電話帳の文字列データを画面表示させる場合、完全カプセル化オブジェクトや非完全カプセル化オブジェクトは、ネイティブプログラムとしてＲＯＭ４０８または不揮発性メモリ４１０に記憶されている表示制御プログラムを使用して液晶画面に電話帳の文字列を表示させる。この表示制御プログラムからJavaＡＰが表示データを取得することができてしまうと、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを用いた意味がなくなってしまう。

【００４８】

しかしながら、ダウンロードされたJavaＡＰが実行される場合には、前述したようにＪＡＭのアクセス制限機能により、JavaＡＰの実行中に携帯電話機４０がアクセスすることのできるリソースが制限される。ここで、JavaＡＰの実行中に携帯電話機４０のアクセスが許可されるリソースに表示制御プログラムは含まれていないので、JavaＡＰが表示制御プログラムから表示データを取得するようなことは一切あり得ない。

【００４９】

また、カプセル化は、プログラミング言語レベルでのカプセル化と、実行コー

ド（マシン語またはバイトコード）レベルでのカプセル化とが考えられる。プログラミング言語レベルでのカプセル化が完全であっても、実行コードレベルでのカプセル化が完全でなければ、データを完全にカプセル化したとは言えない。例えば、プログラミング言語であるC++を用いたプログラムでもprivateフィールドを有するカプセル化オブジェクトを生成することはできる。しかしながら、C++は、プログラミング言語レベルでの完全カプセル化しか達成し得ない。

【００５０】

具体的に説明すると、C++を用いたプログラムにより、オブジェクト内の全てのフィールドをprivateフィールドとして宣言し、カプセル化オブジェクトを生成した場合、このオブジェクト内のprivateフィールドに記憶されているデータを直接読み書きするようなソースコードをコンパイルしようとしてもコンパイルエラーとなり、実行コードは生成されない。

しかしながら、これは、コンパイラによって保証されているに過ぎない。例えば、悪意のある第３者がコンパイラを改造することで、オブジェクト内のprivateフィールドに記憶されているデータを直接読み書きするような実行コードを生成することも可能である。また、コンパイラを改造しなくても、悪意のある第３者がハンドアセンブルなどの手段でオブジェクト内のデータを不正に読み出す実行コードを生成するプログラムを作成することも不可能ではない。加えて、ポインタを用いて直接メモリにアクセスしてしまえば、オブジェクト内のデータを入手することができてしまう。

【００５１】

これに対してJavaの場合、private宣言されたフィールドは、 private属性を有するフィールドであることを示すJavaのバイトコードへコンパイルされる。ＫＶＭがクラスファイルをＲＡＭ４０９などへ展開する際も、フィールドのprivate属性は保持されている。したがって、仮にコンパイラを改造してオブジェクト内のprivateフィールドに記憶されているデータを不正に読み出すようなバイトコードを生成したとしても、ＫＶＭまたはＪＡＭがこれを検知するので、オブジェクト内のデータを入手することはできない。また、Javaはポインタをサポートしていないので、ポインタを用いて直接メモリにアクセスし、オブジェクト内の

データを入手することもできない。

【００５２】

以上のようなことから、Javaでは、プログラミング言語レベルのみに止まらず、バイトコードレベルでの完全なカプセル化を達成することが可能である。

【００５３】

［１－２．実施形態の動作］

次に、本実施形態の動作について説明する。

なお、携帯電話機４０が以下に述べる動作を行う前提として、携帯電話機４０は、移動パケット通信網３０およびインターネット２０を介してコンテンツサーバ１０とパケット通信を行い、コンテンツサーバ１０からJavaＡＰをダウンロードして不揮発性メモリ４１０に記憶しているものとする。また、不揮発性メモリ４１０には、ダウンロードされたJavaＡＰ（ＪＡＲファイルとＡＤＦファイル）の他に、アドレス帳データや電子メールデータ、ユーザデータなどが記憶されているものとする。

【００５４】

＜１－２－１．オブジェクト生成処理＞

まず、携帯電話機４０においてＣＰＵ４０５により実行されるオブジェクト生成処理について図８を参照して説明する。このオブジェクト生成処理は、ＪＡＭの機能としてＣＰＵ４０５により実行されるものであり、例えば、画面表示されたプログラムの一覧リストの中から、実行するプログラムが操作入力により指定された場合などに実行される。なお、プログラムの実行を指示する形態は、操作入力によるものに限定されず、例えば、予め定められた時間毎にプログラムの実行が指示される場合や、既に実行されている他のプログラムから実行が指示される場合、電子メールなどを用いて携帯電話機４０の外部からプログラムの実行が指示される場合などもある。

【００５５】

同図に示すように、まず、携帯電話機４０のＣＰＵ４０５は、実行するプログラムとして操作入力により指定されたプログラムを特定する（ステップＳ１０１）。次いで、ＣＰＵ４０５は、特定したプログラムがダウンロードされたJavaＡ

Ｐであるのか、それともネイティブプログラムであるのかを判別する（ステップＳ１０２）。前述したようにネイティブプログラムには、自身がネイティブプログラムであることを示す識別情報が付与されている。したがって、ＣＰＵ４０５は、プログラムに上記識別情報が付与されているか否かを判別することで、このプログラムがダウンロードされたJavaＡＰであるのか、それともネイティブプログラムであるのかを判別することができる。

【００５６】

その結果、ＣＰＵ４０５は、プログラムがネイティブプログラムであると判別した場合には（ステップＳ１０２：Ｎｏ）、オブジェクト生成処理を終了するとともに、実行するプログラムとして指定されたネイティブプログラムを起動する。そして、ＣＰＵ４０５は、起動させたネイティブプログラムに基づく処理を行なう。

【００５７】

ここで、実行するプログラムがネイティブプログラムである場合は、携帯電話機に記憶されたデータを適法かつ適正に取り扱うことが携帯電話機の製造者により保証されているから、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを用いたり、あるいはネイティブプログラムの実行に伴ってＪＡＭのアクセス制限機能を動作させる必要がない。したがって、ネイティブプログラムが実行される場合、ＪＡＭによるアクセス制限は一切行われず、ネイティブプログラムは、携帯電話機４０内の任意のリソースおよびネットワーク上の任意のリソースにアクセスすることができる。

【００５８】

一方、ＣＰＵ４０５は、プログラムがダウンロードされたJavaＡＰであると判別した場合は（ステップＳ１０２：Ｙｅｓ）、次いで、不揮発性メモリ４１０に記憶されている各種のデータの中から、このJavaＡＰを実行した場合に使用されるデータを、例えば、このJavaＡＰのプログラム内容を解析するなどして特定する（ステップＳ１０３）。なお、JavaＡＰが使用するデータを特定する際には、ＪＡＲストレージ４１０ａ内の、このJavaＡＰのＪＡＲファイルに記憶されているデータは特定の対象から除外する。これは、ＪＡＲファイル内に記憶されてい

るデータは、このJavaＡＰを実行する上で必要となるデータとして当該JavaＡＰを提供するコンテンツプロバイダが用意したデータであるためである。

【００５９】

次いで、ＣＰＵ４０５は、ＡＤＦファイルに内包されているトラステッドアプリケーション識別子を参照して、上記特定したデータを扱うオブジェクトの型を「完全カプセル化」型とするのか、それとも「非完全カプセル化」型とするのかを決定する（ステップＳ１０４）。例えば、ＣＰＵ４０５は、トラステッドアプリケーション識別子が”１”である場合には、当該ＡＤＦファイルに対応するJavaＡＰはトラステッドアプリケーションであるため、上記特定したデータを扱うオブジェクトの型を「完全カプセル化」型に決定する。

【００６０】

この後、ＣＰＵ４０５は、上記ステップＳ１０３において特定したデータと、上記ステップＳ１０４において決定したオブジェクトの型とに基づいて、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成する（ステップＳ１０５）。例えば、上記ステップＳ１０４においてデータを扱うオブジェクトの型を「完全カプセル化」型に決定した場合は、ＣＰＵ４０５は、オリジナル拡張ライブラリ内の完全カプセル化ＡＰＩを起動して、特定されたデータ全てについて完全カプセル化オブジェクトを生成する。また、上記ステップＳ１０４においてデータを扱うオブジェクトの型を「非完全カプセル化」型に決定した場合は、ＣＰＵ４０５は、オリジナル拡張ライブラリ内の非完全カプセル化ＡＰＩを起動して、特定されたデータ全てについて非完全カプセル化オブジェクトを生成する。

【００６１】

次いで、ＣＰＵ４０５は、生成した完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを共通スクラッチパッド４１０ｄに記憶し（ステップＳ１０６）、オブジェクト生成処理を終了する。なお、上記ステップＳ１０５において生成された完全カプセル化オブジェクトや非完全カプセル化オブジェクトは、共通スクラッチパッド４１０ｄではなく、個別スクラッチパッド４１０ｃに記憶される形態であってもよい。

【００６２】

＜１－２－２．アクセス管理処理＞

次に、携帯電話機４０においてＣＰＵ４０５により実行されるアクセス管理処理について図９を参照して説明する。このアクセス管理処理は、ＪＡＭの機能としてＣＰＵ４０５により実行されるものであり、ダウンロードされたJavaＡＰの実行過程においてアクセス要求が発生した場合に、割り込み処理として実行される。

【００６３】

同図に示すように、まず、携帯電話機４０のＣＰＵ４０５は、JavaＡＰの実行過程において発生したアクセス要求について、要求されたアクセス先が予め許可された範囲内のリソースであるか否かを判別し、アクセスを許可するか否かを判定する（ステップＳ２０１）。ここで、アクセスの許可／不許可を判定する仕組みについて具体的に説明すると、ダウンロードされたJavaＡＰが実行される場合、ＣＰＵ４０５は、JavaＡＰの実行に伴ってアクセスすることのできるリソースを、このJavaＡＰのＡＤＦに記述されているＵＲＬにより指定される当該JavaＡＰのダウンロード元のコンテンツサーバ１０と、このJavaＡＰに対して割り当てられたＪＡＲストレージ４１０ｂおよび個別スクラッチパッド４１０ｃ内の記憶領域と、共通スクラッチパッド４１０ｄと、のみに制限する。

【００６４】

したがって、ＣＰＵ４０５は、要求されたアクセス先が上述したリソースのいずれかである場合は、このアクセスを許可する一方、要求されたアクセス先が上述したリソース以外である場合は、このアクセス要求を許可しない。

【００６５】

次いで、ＣＰＵ４０５は、アクセスの許可／不許可を示す判定結果を要求元のJavaＡＰに通知した後（ステップＳ２０２）、アクセス管理処理を終了する。また、実行中のJavaＡＰは、ＪＡＭによる判定結果を受け取ると、この判定結果に従って、アクセスが許可された場合は当該アクセス要求に基づく処理を実行する一方、アクセスが許可されなかった場合は当該アクセス要求に基づく処理をキャンセルする。

【００６６】

さて、携帯電話機４０のＣＰＵ４０５は、ダウンロードしたJavaＡＰを実行する場合、図８に示したオブジェクト生成処理を行った後にJavaＡＰを起動する。また、ダウンロードしたJavaＡＰの実行過程においてＣＰＵ４０５は、アクセス要求が発生すると、図９に示したアクセス管理処理を行う。したがって、携帯電話機４０は、ダウンロードしたJavaＡＰの実行中において必ずＪＡＭによるアクセス制限を受けることとなり、例えば、不揮発性メモリ４１０に記憶されているアドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータ、コンテンツなどのデータそのものにアクセスすることができなくなる。

【００６７】

このため、携帯電話機４０のＣＰＵ４０５は、上述したオブジェクト生成処理において、起動させるJavaＡＰが使用するデータを特定し、当該データ用の完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成して共通スクラッチパッド４１０ｄに記憶する。この共通スクラッチパッド４１０ｄは、前述したように、ＪＡＭによるアクセス制限が行われている場合であっても携帯電話機４０のアクセスが許可されるリソースである。また、携帯電話機４０にダウンロードされるJavaＡＰは、共通スクラッチパッド４１０ｄに記憶された完全カプセル化オブジェクトや非完全カプセル化オブジェクトにアクセスし、当該オブジェクトに備わるメソッドを使用してこのオブジェクト内のデータに対する操作を指令するように作成されている。

【００６８】

例えば、アドレス帳データを使用する非トラステッドアプリケーションが起動される場合、上述したオブジェクト生成処理によりアドレス帳データ用の完全カプセル化オブジェクトが生成され、共通スクラッチパッド４１０ｄに記憶される。また、この非トラステッドアプリケーションは、上記生成されたアドレス帳データ用の完全カプセル化オブジェクトに対して、当該オブジェクトに備わるメソッドを用いてこのオブジェクト内のデータに対する操作を指令する。したがって、完全カプセル化オブジェクトの有するアドレス帳データの一部を画面表示させることなどが可能となる一方、完全カプセル化オブジェクトの有するデータその

ものが非トラステッドアプリケーションに引き渡されることはない。

【００６９】

従来は、ダウンロードされたJavaＡＰに対するセキュリティを確保するため、このようなJavaＡＰについては、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータなどに一切アクセスすることができなかった。これに対して本実施形態によれば、完全カプセル化オブジェクトを用いることにより、データそのものがJavaＡＰに引き渡されることがないので、ダウンロードされたJavaＡＰに対するセキュリティを確保しつつ、同時に、従来は一切アクセスできなかったデータについて、完全カプセル化オブジェクトを介して画面表示を行わせることなどができるようになる。したがって、ダウンロードされたJavaＡＰが携帯電話機４０において実現することのできる機能を充実させることができる。

また、本実施形態によれば、JavaAPを作成するプログラマは、オブジェクトを使用してデータにアクセスするプログラムをコーディングすることができるために、データへのアクセス方法やセキュリティを考慮せずにプログラムのコーディングを行うことができる。これにより、プログラマの開発生産性や保守性が向上する。

【００７０】

＜１－２－３．JavaＡＰ終了処理＞

次に、携帯電話機４０においてＣＰＵ４０５により実行されるJavaＡＰ終了処理について図１０を参照して説明する。このJavaＡＰ終了処理は、ＪＡＭの機能としてＣＰＵ４０５により実行されるものであり、JavaＡＰの実行終了要求が発生した場合に、割込処理として実行される。

【００７１】

同図に示すように、携帯電話機４０のＣＰＵ４０５は、JavaＡＰの実行終了要求が発生すると、共通スクラッチパッド４１０ｄに記憶されている完全カプセル化オブジェクトや非完全カプセル化オブジェクトを削除する（ステップＳ３０１）。このステップＳ３０１において削除される完全カプセル化オブジェクトや非完全カプセル化オブジェクトは、JavaＡＰを起動させる際に、上述したオブジェ

クト生成処理（図８参照）において生成され、共通スクラッチパッド４１０ｄに記憶されたものである。ＣＰＵ４０５は、共通スクラッチパッド４１０ｄからオブジェクトを削除すると、JavaＡＰ終了処理を終える。

【００７２】

このようにダウンロードされたJavaＡＰを起動する際に、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを生成して共通スクラッチパッド４１０ｄに記憶する一方、このJavaＡＰの実行が終了する際に、共通スクラッチパッド４１０ｄから完全カプセル化オブジェクトや非完全カプセル化オブジェクトを削除するようにすれば、携帯電話機４０のメモリ資源を効率的に活用することができる。

【００７３】

［２．第２実施形態］

上記第１実施形態においては、JavaＡＰがトラステッドアプリケーションか否かによって、データの種類に関わらず一律に完全カプセル化オブジェクトを生成するか、非完全カプセル化オブジェクトを生成するかを決定したが、この第２実施形態においては、JavaＡＰに化体している信頼のレベルとデータの保護に求められる確実さを示す重要度とによって、完全カプセル化オブジェクトを生成するか、非完全カプセル化オブジェクトを生成するかを決定する。さらに、JavaＡＰの信頼のレベルによって、使用可能なオブジェクトを決定する。

【００７４】

［２－１.実施形態の構成］

第１実施形態における、ＡＤＦの「トラステッドアプリケーション識別子」の値は、一定の信頼が化体していないJavaＡＰの場合には“０”、一定の信頼が化体しているJavaＡＰの場合には“１”が設定されていたが、第２実施形態においては、JavaＡＰに化体している信頼のレベル（以下、「JavaＡＰの信頼度」という）に応じて、“高”、“中”、“低”のレベルが設定される。ここで、例えば、JavaＡＰに化体している信頼度が高いとは、JavaＡＰによってデータが適正に取り扱われる確率が予め定められた基準確率より高いことを意味する。

【００７５】

＜２－１－１．携帯電話機の構成＞

携帯電話機４０の不揮発性メモリ４１０には、図１１に示すように、重要度テーブル４１０ｄが設けられている。

同図に示すように、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータのような、携帯電話機４０に記憶されているデータの中でも特にデータを確実に保護する必要が求められているデータには、重要度の値が“高”に設定される。また、求められる保護の確実さが中間のデータには重要度の値が“中”、低いデータには重要度の値が“低”に設定される。

また、携帯電話機４０の不揮発性メモリ４１０には、図１２に示すように、アプリデータ関係テーブル４１０ｅが設けられている。このテーブルには、データの重要度とJavaＡＰの信頼度との組み合わせによって、当該データを完全カプセル化型として扱うのか、それとも非完全カプセル化型として扱うのかが設定されている。例えば、同図において、信頼度が高いJavaＡＰの場合は、データの重要度に関わらず非カプセル化型が設定されている。また、信頼度が低いJavaＡＰの場合は、データの重要度が“高”および“中”の場合に完全カプセル化型が設定され、データの重要度が“低”の場合に非完全カプセル化型が設定されている。

また、このアプリデータ関係テーブル４１０ｅには、JavaＡＰの信頼度によって、使用可能なオブジェクトが設定されている。例えば、同図において、信頼度が高いJavaＡＰは、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトを使用可能である。よって、JavaＡＰは、JavaＡＰ実行時に、データの重要度に関わらず生成される非完全カプセル化オブジェクトを使用することができる。また、信頼度が低いJavaＡＰも、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトを使用可能である。よって、JavaＡＰは、JavaＡＰ実行時に、データの重要度が“高”または“中”の場合に生成される完全カプセル化オブジェクトとデータの重要度が“低”の場合に生成される非完全カプセル化オブジェクトとを使用することができる。

なお、以上説明した重要度テーブル４１０ｄおよびアプリデータ関係テーブル４１０ｅの内容は、携帯電話機４０出荷時に予め登録されているが、サーバからダウンロードされるコンテンツは、ダウンロード時に重要度テーブル４１０ｄに

データが記憶される。また、ユーザが携帯電話機４０を操作して、これらのテーブルに値を入力することが可能である。

以上説明した構成以外の構成は、第１実施形態と同様であるため、重複した説明を省略する。

【００７６】

［２－２．実施形態の動作］

次に、本実施形態の動作について説明する。

不揮発性メモリ４１０には、ダウンロードされたJavaＡＰの他に、アドレス帳データや電子メールデータ、ユーザデータなどが記憶されており、重要度テーブル４１０ｄおよびアプリデータ関係テーブル４１０ｅには図１１および図１２の情報内容が記憶されているものとする。

【００７７】

＜２－２－１．オブジェクト生成処理＞

次に、オブジェクト生成処理を、第１実施形態における図８に示すフローチャートを用いて説明する。

ステップＳ１０１～ステップＳ１０３までは、第１実施形態と同様である。

次に、ＣＰＵ４０５は、不揮発性メモリ４１０よりJavaＡＰに対応するＡＤＦのトラステッド識別子を参照して、JavaＡＰの信頼度を取得する。次いで、ＣＰＵ４０５は、重要度テーブル４１０ｄを参照して、ステップＳ１０３で特定したデータの重要度を取得する。次いで、ＣＰＵ４０５は、アプリデータ関係テーブル４１０ｅを参照して、データの重要度とJavaＡＰの信頼度とに基づいて、このデータを扱うオブジェクトの型を「完全カプセル化」型とするのか、それとも「非完全カプセル化」型とするのかを決定する（ステップＳ１０４）。例えば、JavaＡＰの使用するデータがアドレス帳データの場合、ＣＰＵ４０５は、重要度テーブル４１０ｄから、アドレス帳データの重要度“高”を読み出す。また、例えば、JavaＡＰに対応するＡＤＦのトラステッド識別子から取得した信頼度が“低”である場合、アプリデータ関係テーブル４１０ｅより、このアドレス帳データを扱うオブジェクトの型を「完全カプセル化」型に決定する。

【００７８】

この後、ＣＰＵ４０５は、上記ステップＳ１０３において特定したデータと、上記ステップＳ１０４において決定したオブジェクトの型とに基づいて、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成する（ステップＳ１０５）。上記ステップＳ１０３において決定したオブジェクト型が完全カプセル化型の場合、ＣＰＵ４０５は、オリジナル拡張ライブラリ内の完全カプセル化ＡＰＩを起動して、完全カプセル化オブジェクトを生成する。また、上記ステップＳ１０３において決定したオブジェクト型が非完全カプセル化型の場合、ＣＰＵ４０５は、オリジナル拡張ライブラリ内の非完全カプセル化ＡＰＩを起動して、非完全カプセル化オブジェクトを生成する。

次いで、ＣＰＵ４０５は、生成した完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを共通スクラッチパッド４１０ｄに記憶し（ステップＳ１０６）、オブジェクト生成処理を終了する。

【００７９】

また、上記ステップＳ１０３においてJavaＡＰの使用するデータが複数特定された場合は、特定した各データ毎に、当該データ用の完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成して共通スクラッチパッド４１０ｄに記憶するため、上記ステップＳ１０４～Ｓ１０６までの処理を各データ毎に繰り返して行う。そして、ＣＰＵ４０５は、オブジェクト生成処理を終了した後、実行するプログラムとして指定されたJavaＡＰを起動し、このプログラムに基づく処理を開始する。

【００８０】

＜２－２－２．オブジェクト使用管理処理＞

次に、携帯電話機４０においてＣＰＵ４０５により実行されるオブジェクト使用管理処理について図１３を参照して説明する。

【００８１】

同図に示すように、まず、携帯電話機４０のＣＰＵ４０５は、JavaＡＰの実行過程においてオブジェクトを使用するための要求が発生した際、当該オブジェクトがJavaＡＰに使用が許可されているか否かをアプリデータ関係テーブル４１０ｅを参照することにより判別し、オブジェクトの使用を許可するか否かを判定す

る（ステップＳ４０１）。ここでは、図１２のアプリデータ関係テーブル４１０ｅに示す通り、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトの双方とも使用可能であるため、オブジェクトの使用は許可される。

【００８２】

次いで、ＣＰＵ４０５は、オブジェクト使用の許可／不許可を示す判定結果を要求元のJavaＡＰに通知した後（ステップＳ４０２）、オブジェクト使用管理処理を終了する。また、実行中のJavaＡＰは、判定結果を受け取ると、この判定結果に従って、使用が許可された場合は当該使用するための要求に基づく処理を実行する一方、使用が許可されなかった場合は当該使用するための要求に基づく処理をキャンセルする。

ここでは、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトとも使用が許可され、JavaＡＰは、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを使用して、データにアクセスする。

【００８３】

上述した動作以外の動作（アクセス管理処理、JavaＡＰ終了処理）は、第１実施形態と同様であるため、重複した説明を省略する。

【００８４】

以上のように、JavaＡＰの信頼度とデータの重要度とによって、生成するオブジェクトを決定し、さらに、JavaＡＰの信頼度によって使用可能なオブジェクトを決定しているため、データとJavaＡＰとの様々な組合わせに応じて、JavaＡＰがデータに対してアクセスすることのできる許可レベル（データ自体をJavaＡＰに引き渡す、引き渡さず取得するのみ、取得もしない）を設定することができる。このため、データに対するセキュリティを保持しながら、様々なJavaＡＰを携帯電話機４０で動作させることが可能となる。また、この許可レベルの設定は、重要度テーブル４１０ｄおよびアプリデータ関係テーブル４１０ｅを使用して行うことができる。そして、JavaＡＰを作成するプログラマは、データにアクセスするプログラムをコーディングする際に、セキュリティやデータへのアクセス方法を考慮せずに、これらのテーブルによって許可レベルが予め設定されているオブジェクトを使用すればよい。このため、プログラマの開発生産性や保守性が向

上する。

【００８５】

［３．変形例］

以上、本発明の実施形態について説明したが、本発明はその主要な特徴から逸脱することなく他の様々な形態で実施することが可能である。上述した実施形態は、本発明の一態様を例示したものに過ぎず、本発明の範囲は、特許請求の範囲に示す通りであって、また、特許請求の範囲の均等範囲に属する変形や変更は、全て本発明の範囲内に含まれる。なお、変形例としては、例えば、以下のようなものが考えられる。

【００８６】

（１）上記第２実施形態において使用したアプリデータ関係テーブル４１０ｅに設定した内容は、一例に過ぎず、例えば、図１４に示すような内容をアプリデータ関係テーブル４１０ｅに設定することもできる。

この場合、ＣＰＵ４０５は、このアプリデータ関係テーブル４１０ｅを参照することによって、トラステッドアプリケーション識別子の信頼度に関わらずデータの重要度が“高”の場合に、完全カプセル化オブジェクトを生成し、データの重要度が“中”および“低”の場合に、非完全カプセル化オブジェクトを生成する。そして、ＪａｖａＡＰは、ＪａｖａＡＰ実行の際に、ＪａｖａＡＰに対応するトラステッドアプリケーション識別子の信頼度が“低”である場合には、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクト共に使用せず、トラステッドアプリケーション識別子の信頼度が“中”である場合には、完全カプセル化オブジェクトのみを使用し、トラステッドアプリケーション識別子の信頼度が“高”である場合には、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトの双方を使用する。このようにすることで、ＣＰＵ４０５は、生成するオブジェクトをデータの重要度のみで決定することができ、生成したオブジェクトを使用する際には、ＪａｖａＡＰに化体した信頼のレベルのみによって、生成したオブジェクトを使用するか否かを決定することができる。

【００８７】

（２）上述した第２実施形態では、重要度テーブル４１０ｄとアプリデータ関係

テーブル４１０ｅを用いる構成としたが、これらのテーブルは、データ構成の概念を示すものに過ぎず、通信装置に応じて最適なデータ構成を選択することができる。例えば、アドレス帳データ、電子メールデータ、コンテンツなどのデータ自体に重要度を示すデータを付与するようにすれば、重要度テーブル４１０ｄを用いる必要はない。

【００８８】

（３）上記実施形態においては、ダウンロードされたJavaＡＰが、トラステッドアプリケーションか否かを識別したり、JavaＡＰの信頼度を識別するのに、JavaＡＰに対応するＡＤＦに内包されているトラステッドアプリケーション識別子を用いたが、これは一例に過ぎない。例えば、JavaＡＰの信頼度や、JavaＡＰがトラステッドアプリケーションとして認定済である、等のJavaＡＰに化体した信頼に関するデータを管理するための管理サーバ装置を設置して、携帯電話機４０のＣＰＵ４０５が、実行しようとするJavaＡＰに化体した信頼に関するデータが管理サーバに存在している場合には、当該データを管理サーバより受信するようにしてもよい。

また、上記実施形態においては、インターネット２０に接続されたコンテンツサーバ１０よりダウンロードされたJavaＡＰを用いた場合について説明したが、本発明は、ネイティブプログラム以外の任意のプログラム、すなわち携帯電話機４０の販売後にそのメモリに記憶されるプログラムに用いた場合にも効果を奏する。例えば、赤外線インターフェースを備えた携帯電話機４０が、赤外線インターフェースを備えたパーソナルコンピュータ等の通信機器より赤外線通信によってプログラムを受信し、そのプログラムに化体している信頼に関するデータを管理サーバから受信するようにしてもよい。

【００８９】

（４）上述した実施形態では、ダウンロードされたJavaＡＰの実行が指示された場合に、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成するようにしたが、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトが生成されるタイミングは、JavaＡＰの実行が指示されたときに限定されるものではない。例えば、JavaＡＰがデータを参照するタイミングでオブジェ

クトを生成してもよい。

【００９０】

（５）上述した実施形態において、コンテンツサーバ１０は、インターネット２０に接続されている構成とした。しかしながら、コンテンツサーバ１０は、専用線を介して移動パケット通信網３０のゲートウェイサーバ３１に直接接続されている構成であってもよい。

【００９１】

（６）上述した実施形態では、図１５においてハッチングで示すように、ＫＶＭと、コンフィギレーションとしてＣＬＤＣを備えるとともにプロファイルとしてオリジナルJava拡張プロファイルを備えるＪ２ＭＥとが記憶された携帯電話機４０に本発明を適用した場合について説明した。しかしながら、Java実行環境は、上述したＫＶＭとＪ２ＭＥの組み合わせに限定されるものではない。また、本発明が適用可能な通信装置は、携帯電話機に限定されるものではない。

例えば、同図に示すように、Ｊ２ＭＥのプロファイルとして、オリジナルJava拡張プロファイルの代わりにＭＩＤＰ（Mobile Information Device Profile）を有する構成であってもよい。また、ＫＶＭの代わりにＪＶＭを有し、Ｊ２ＭＥのコンフィギレーションとして ＣＬＤＣの代わりにＣＤＣ（Connected Device Configuration）を、また、Ｊ２ＭＥのプロファイルとして、例えば、液晶付電話機用プロファイル、ＴＶ用プロファイル、カーナビゲーション用プロファイルなどを有する構成であってもよい。さらには、ＨｏｔＳｐｏｔと、Ｊ２ＳＥ（Java 2 Standard Edition）またはＪ２ＥＥ（Java 2 Enterprise Edition）とを有する構成であってもよい。

【００９２】

（７）また、以上説明したJava実行環境の変形例から明らかなように、本発明は、例えば、ＰＨＳ（Personal Handyphone System：登録商標）端末やＰＤＡ（Personal Digital Assistant）、カーナビゲーション装置、パーソナルコンピュータなどの、通信機能を有する各種電子機器に適用可能である。また、本発明は、移動パケット通信網３０に収容される通信装置に限定されるものではない。例えば、図１６に示すような通信システム２において、ＬＡＮ５０内に設けられたパ

ーソナルコンピュータ７０Ａ～７０Ｃに本発明を適用することもできる。

【００９３】

（８）また、上述した実施形態では、Javaプログラミング言語により記述されたJavaＡＰを用いた場合について説明したが、プログラミング言語はJavaに限定されるものではない。例えば、システムに要求されるセキュリティのレベルによっては、、Ｃ++を使用してシステムを構築することも可能である。

【００９４】

【発明の効果】

以上説明したように、本発明によれば、ダウンロードされるプログラムに化体した信頼のレベルとデータの重要度に応じて、通信装置に記憶されているデータに対するアクセス制御を行うことができるため、データに対するセキュリティを確保しながら様々なプログラムによってデータを操作することができる。

【図面の簡単な説明】

【図１】　本発明の第１実施形態に係る通信システムの構成を示すブロック図である。

【図２】　同実施形態に係る携帯電話機のハードウェア構成を示すブロック図である。

【図３】　同実施形態に係る携帯電話機において、不揮発性メモリに記憶されている型ＡＤＦファイルのデータ構成を示す図である。

【図４】　同実施形態に係る携帯電話機において、JavaＡＰの実行環境を説明するための図である。

【図５】　同実施形態に係る携帯電話機において、カプセル化オブジェクトを説明するための模式図である。

【図６】　同実施形態に係る携帯電話機において、非完全カプセル化オブジェクトについて例示する模式図である。

【図７】　同実施形態に係る携帯電話機において、完全カプセル化オブジェクトについて例示する模式図である。

【図８】　同実施形態に係る携帯電話機において、ＣＰＵにより実行されるオブジェクト生成処理の動作を説明するフローチャートである。

【図９】　同実施形態に係る携帯電話機において、ＣＰＵにより実行されるアクセス管理処理の動作を説明するフローチャートである。

【図１０】　同実施形態に係る携帯電話機において、ＣＰＵにより実行されるJavaＡＰ終了処理の動作を説明するフローチャートである。

【図１１】　本発明の第２実施形態に係る携帯電話機において、不揮発性メモリに記憶されている重要度テーブルのデータ構成を示す図である。

【図１２】　同実施形態に係る携帯電話機において、不揮発性メモリに記憶されているアプリデータ関係テーブルのデータ構成を示す図である。

【図１３】　同実施形態に係る携帯電話機において、ＣＰＵにより実行されるオブジェクト使用管理処理の動作を説明するフローチャートである。

【図１４】　本発明の変形例に係る携帯電話機において、不揮発性メモリに記憶されているアプリデータ関係テーブルのデータ構成を示す図である。

【図１５】　本発明の変形例に係り、Java実行環境の変形例を説明するための図である。

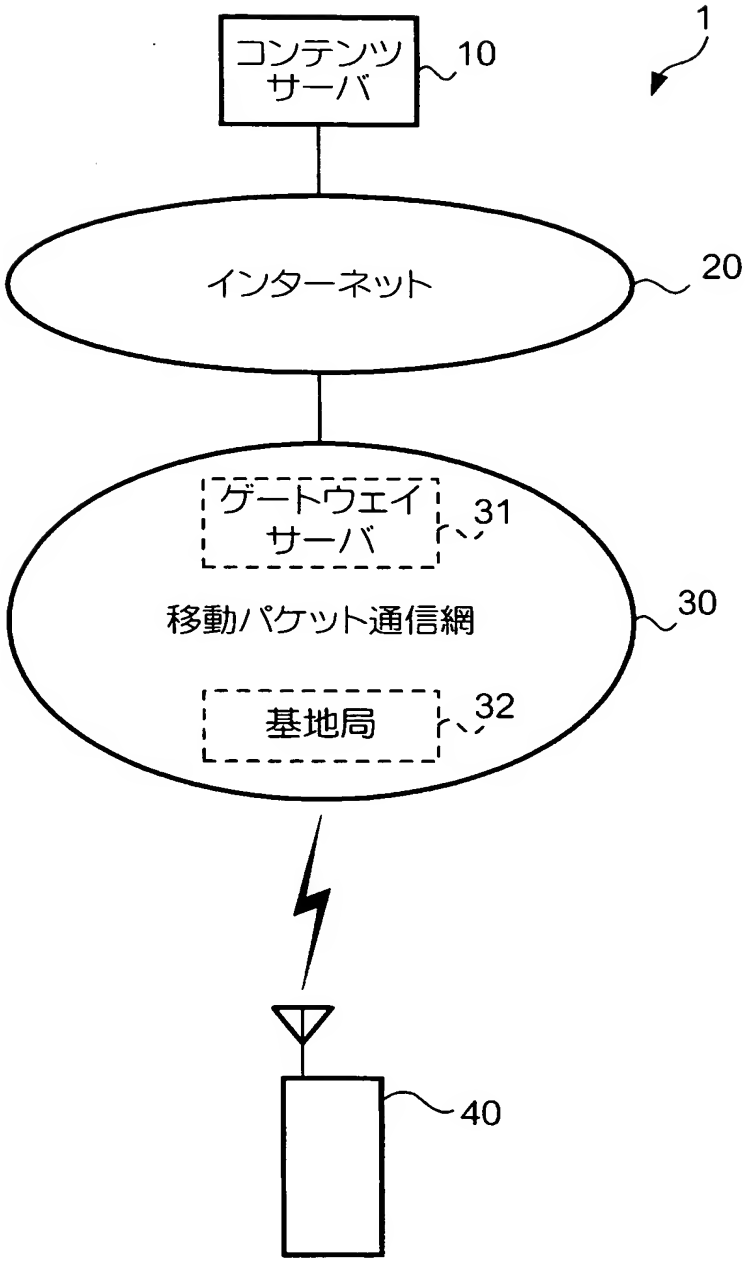
【図１６】　本発明の変形例に係り、通信システムの変形例を例示するブロック図である。

【符号の説明】

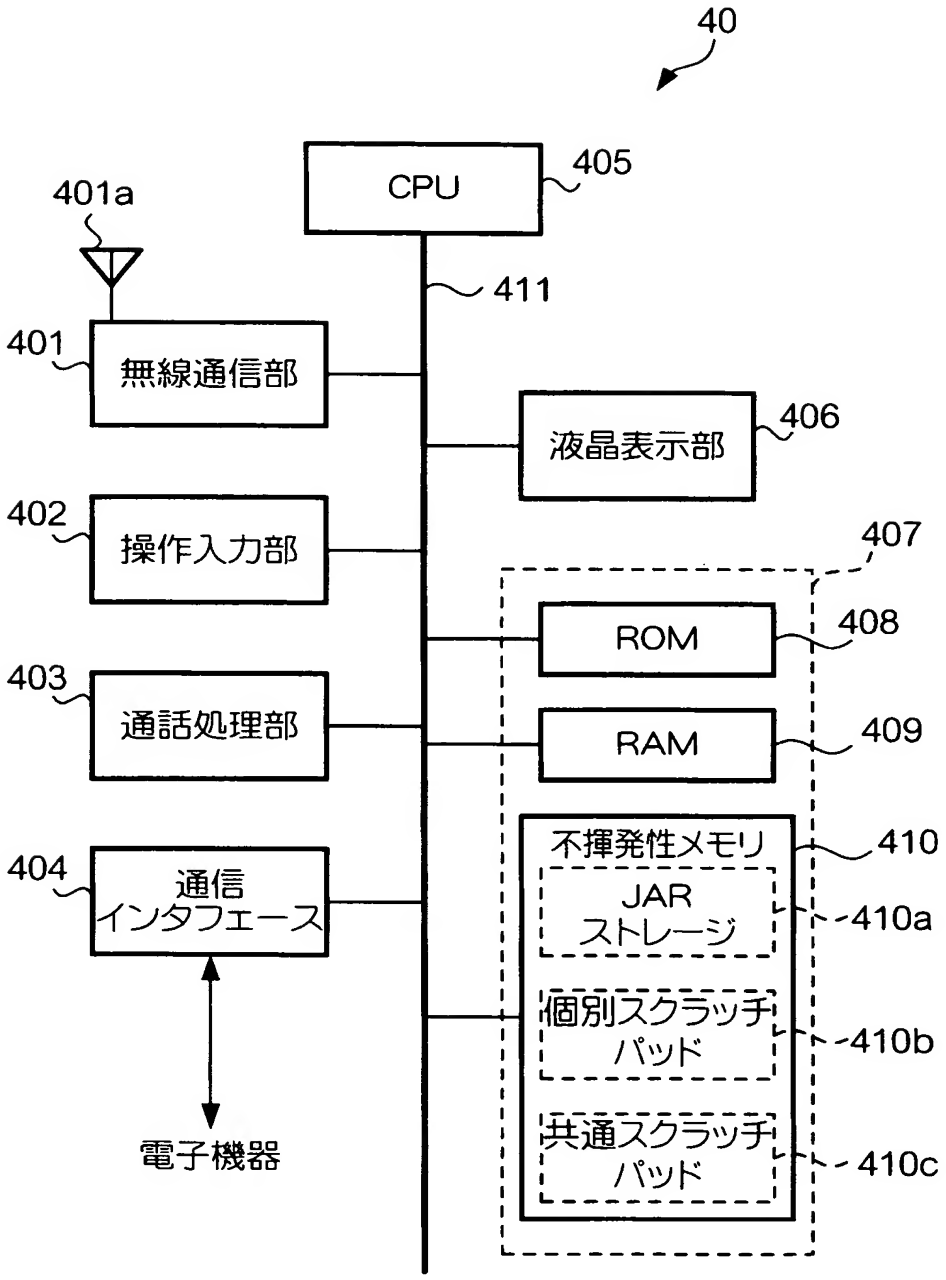
１，２……通信システム、１０……コンテンツサーバ、２０……インターネット、３０……移動パケット通信網、３１……ゲートウェイサーバ、３２……基地局、４０……携帯電話機、５０……ＬＡＮ、６０……ゲートウェイサーバ、７０ａ，７０ｂ，７０ｃ……パーソナルコンピュータ、４０１……無線通信部、４０１ａ……アンテナ、４０２……操作入力部、４０３……通話処理部、４０４……通信インタフェース、４０５……ＣＰＵ、４０６……液晶表示部、４０７……記憶部、４０８……ＲＯＭ、４０９……ＲＡＭ、４１０……不揮発性メモリ、４１０ａ……ＪＡＲストレージ、４１０ｂ……個別スクラッチパッド、４１０ｃ……共通スクラッチパッド、４１０ｄ……重要度テーブル、４１０ｅ……アプリデータ関係テーブル。

【書類名】　図面

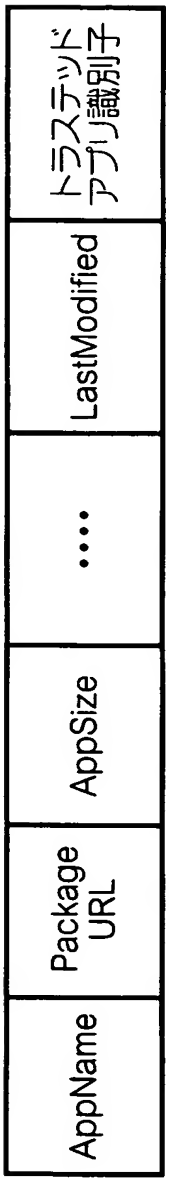
【図１】



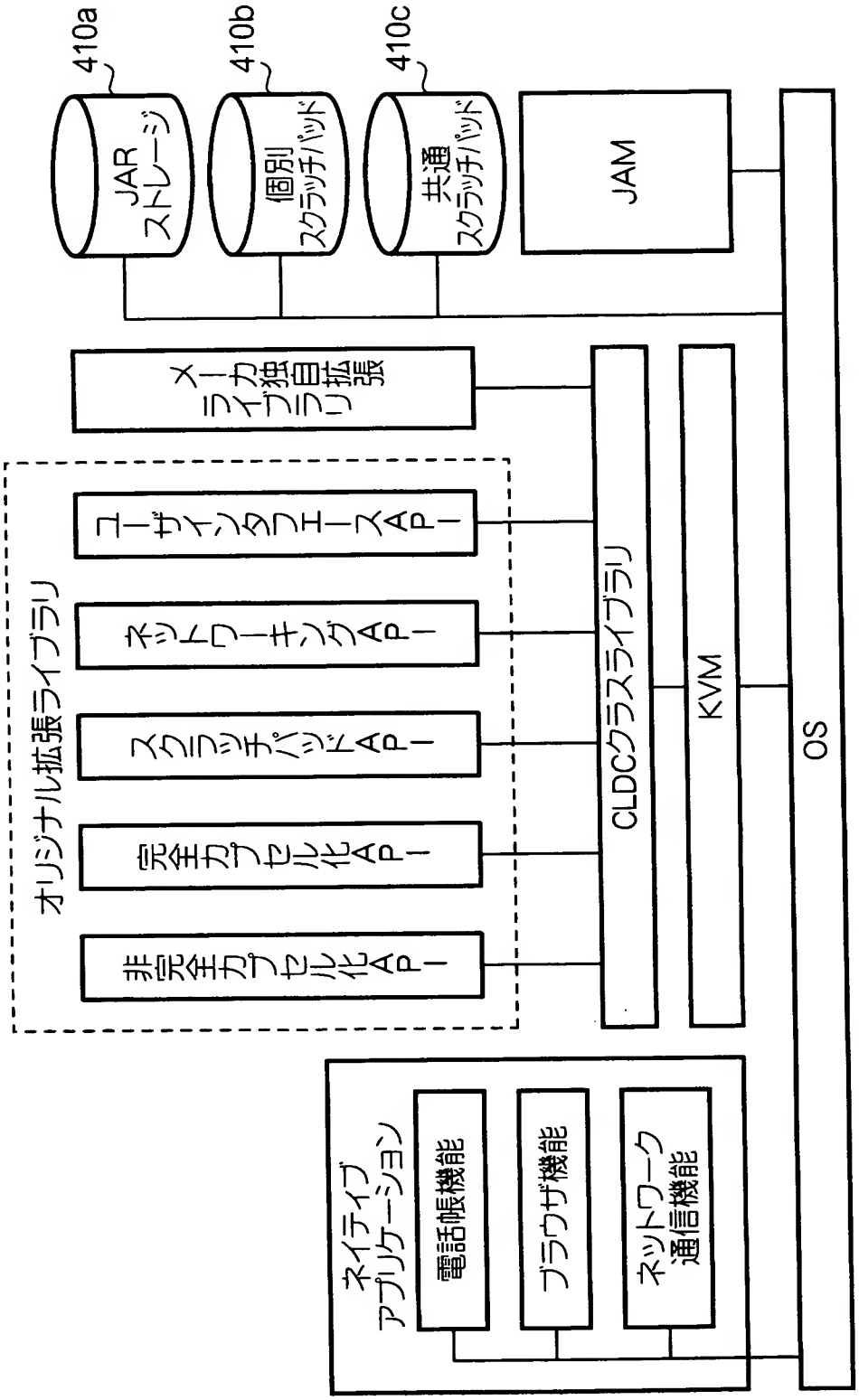
【図２】


40
CPU
405
401a
411
401
無線通信部
液晶表示部
406
402
操作入力部
407
ROM
408
403
通話処理部
RAM
409
不揮発性メモリ
410
404
通信
インタフェース
JAR
ストレージ
410a
個別スクラッチ
パッド
410b
共通スクラッチ
パッド
410c
電子機器

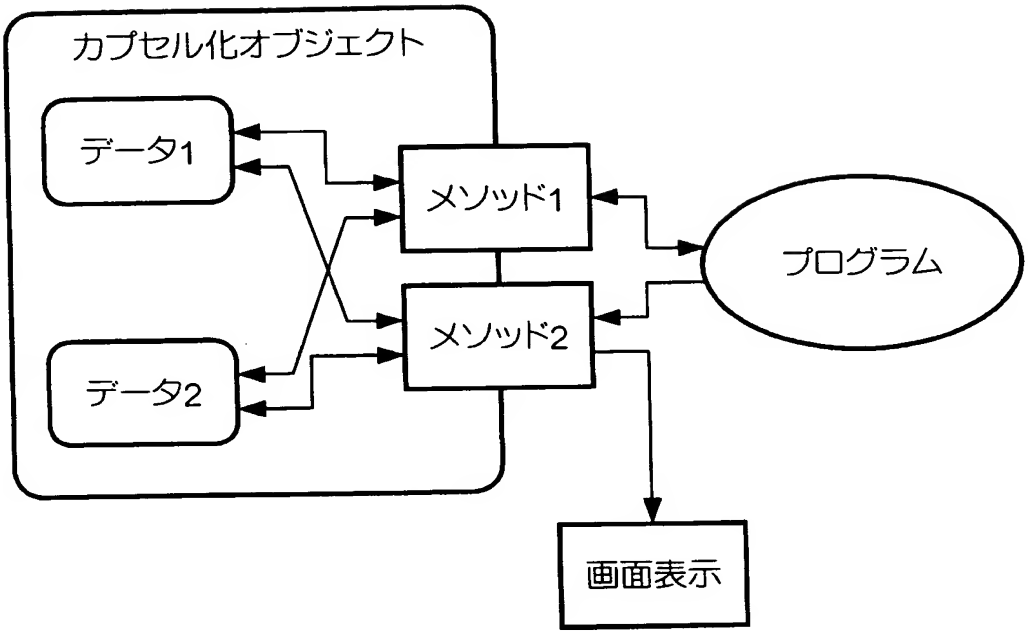
【図３】



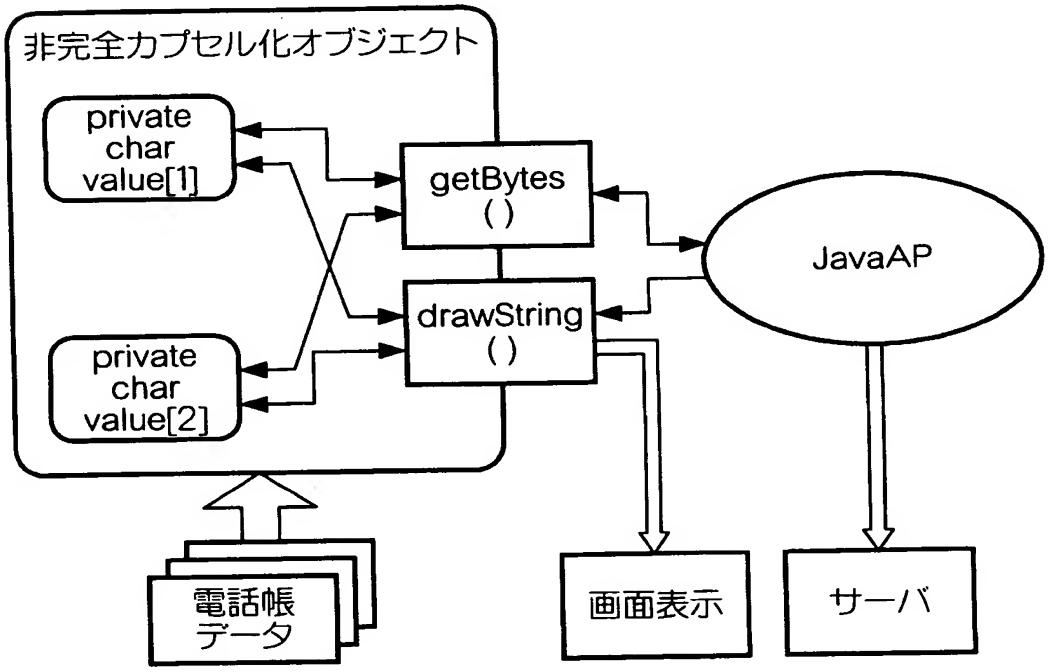
【図４】


410a
410b
410c
JARストレージ
個別スクラッチパッド
共通スクラッチパッド
JAM
メーカ独自ライブラリ拡張
オリジナル拡張ライブラリ
ユーザインタフェースAPI
ネットワーキングAPI
スクラッチパッドAPI
完全カプセル化API
非完全カプセル化API
CLDCクラスライブラリ
KVM
OS
ネイティブアプリケーション
電話帳機能
ブラウザ機能
ネットワーク通信機能

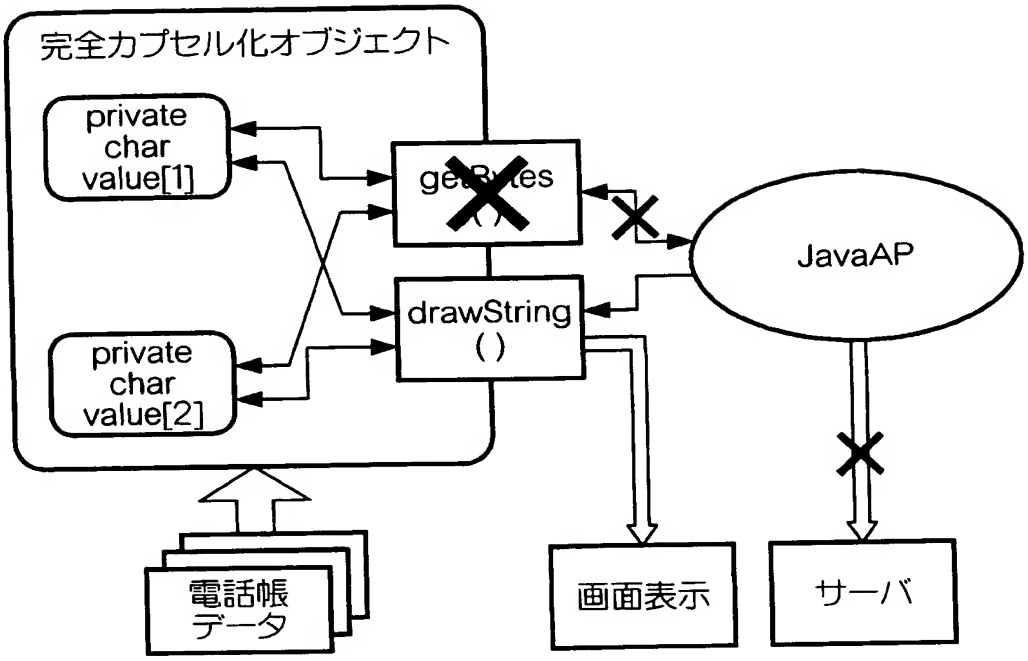
【図５】


カプセル化オブジェクト
データ1
データ2
メソッド1
メソッド2
プログラム
画面表示

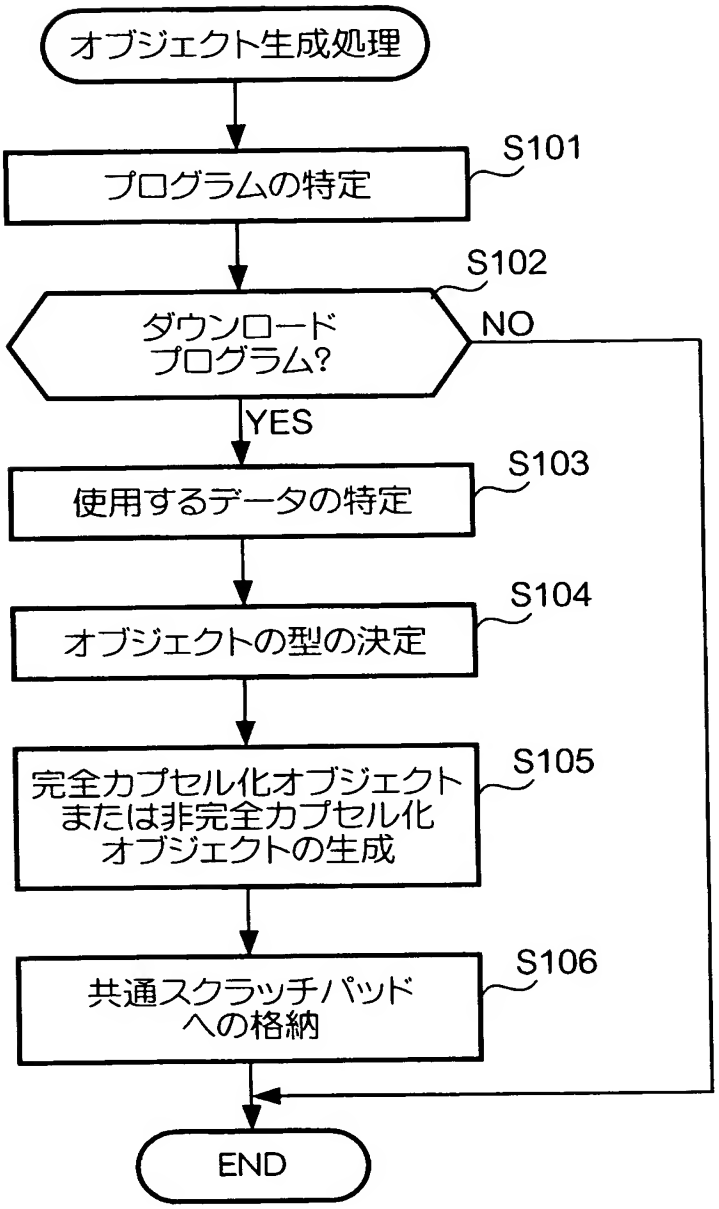
【図６】


非完全カプセル化オブジェクト
private char value[1]
private char value[2]
getBytes ( )
drawString ( )
JavaAP
電話帳データ
画面表示
サーバ

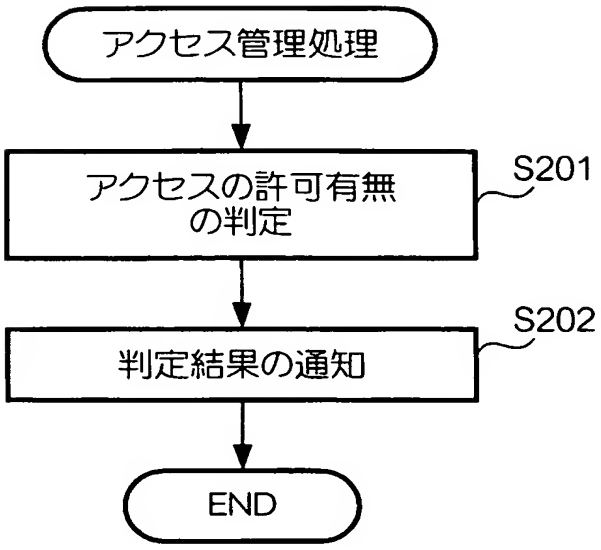
【図７】


完全カプセル化オブジェクト
private
char
value[1]
getBytes
( )
JavaAP
drawString
( )
private
char
value[2]
電話帳
データ
画面表示
サーバ

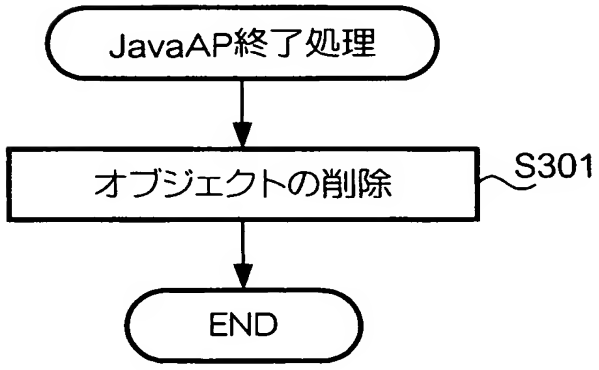
【図８】


オブジェクト生成処理
プログラムの特定
S101
ダウンロード
プログラム?
S102
NO
YES
使用するデータの特定
S103
オブジェクトの型の決定
S104
完全カプセル化オブジェクト
または非完全カプセル化
オブジェクトの生成
S105
共通スクラッチパッド
への格納
S106
END

【図９】


アクセス管理処理
アクセスの許可有無の判定
S201
判定結果の通知
S202
END

【図１０】


JavaAP終了処理
オブジェクトの削除
S301
END

【図１１】

410d

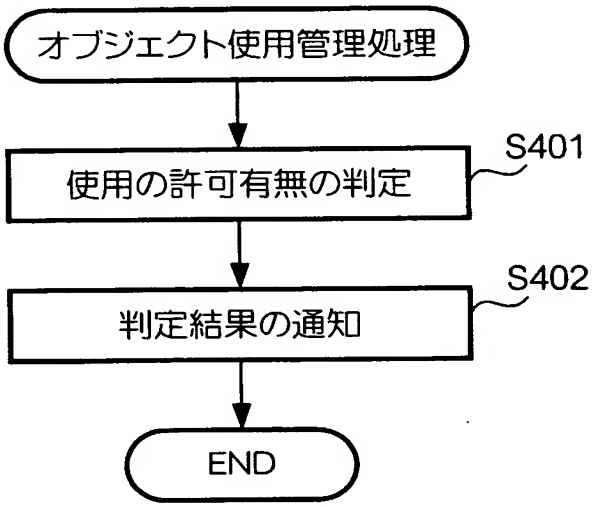
データ名	重要度
アドレス帳データ	高
電子メールデータ	高
着信・発信履歴データ	高
ユーザデータ	高
コンテンツA	中
コンテンツB	低
自作画像データ	低
……	……

【図１２】

410e

		トラステッドアプリ識別子		
		信頼度 低	信頼度 中	信頼度 高
データ	重要度 高	完全 カプセル化型	完全 カプセル化型	非完全 カプセル化型
	重要度 中	完全 カプセル化型	非完全 カプセル化型	非完全 カプセル化型
	重要度 低	非完全 カプセル化型	非完全 カプセル化型	非完全 カプセル化型
使用可能な オブジェクト		完全カプセル化 オブジェクト、 非完全カプセル 化オブジェクト	完全カプセル化 オブジェクト、 非完全カプセル 化オブジェクト	完全カプセル化 オブジェクト、 非完全カプセル 化オブジェクト

【図１３】

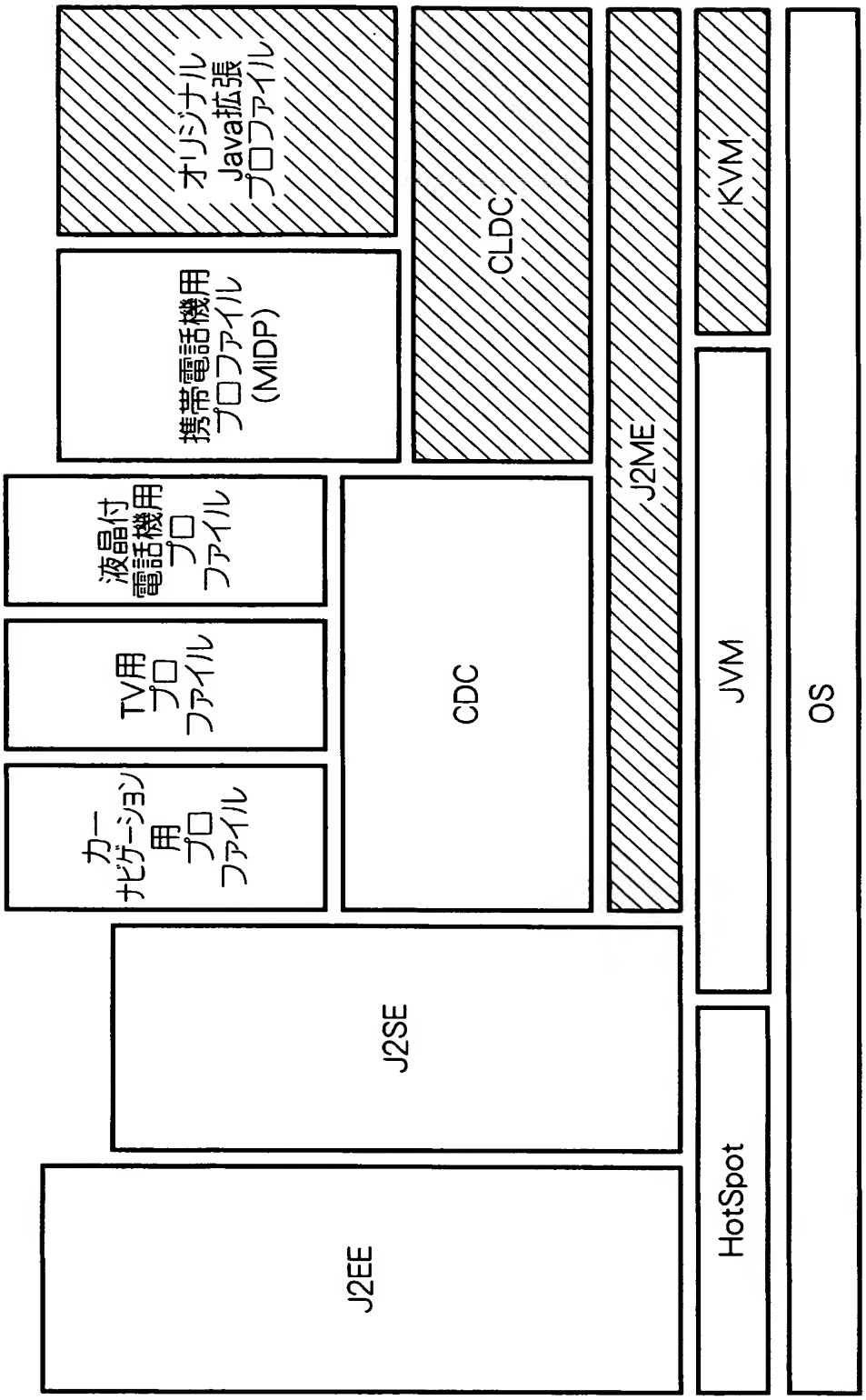


【図１４】

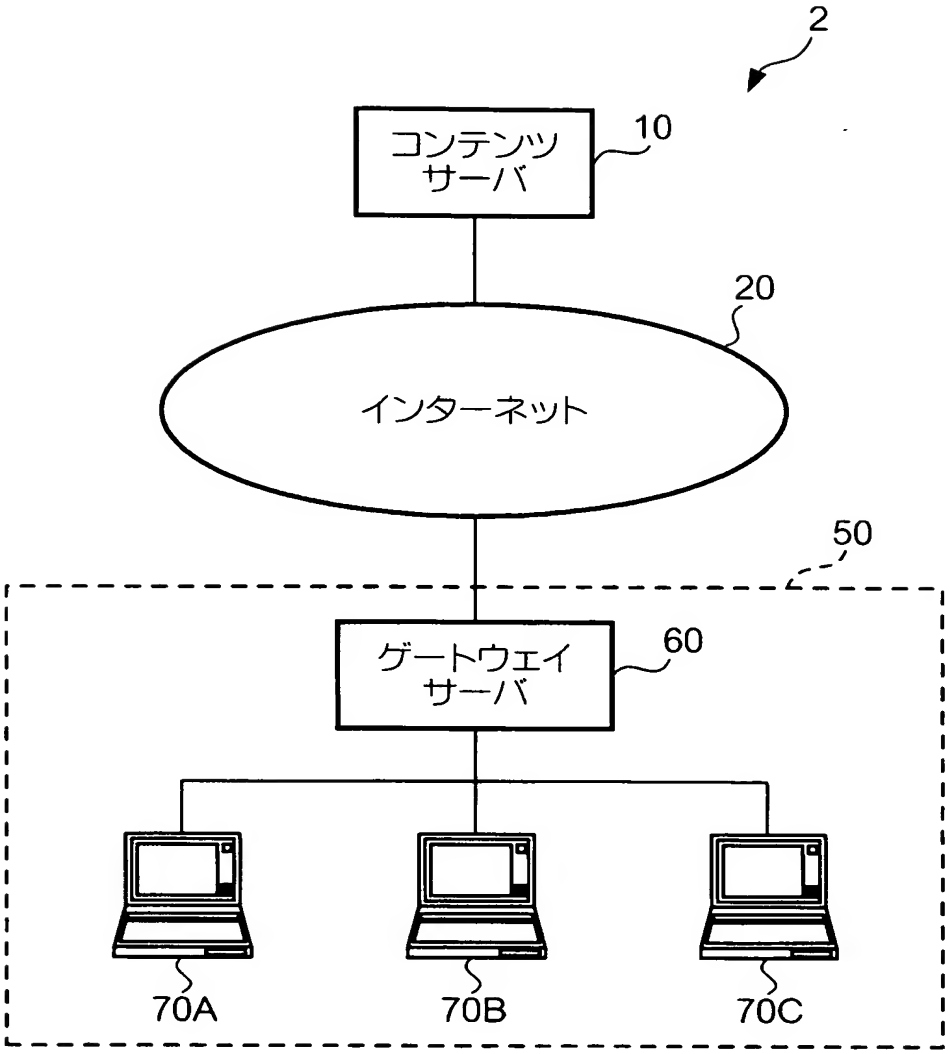
410e

		トラステッドアプリ識別子		
		信頼度 低	信頼度 中	信頼度 高
データ	重要度 高	完全 カプセル化型	完全 カプセル化型	完全 カプセル化型
	重要度 中	非完全 カプセル化型	非完全 カプセル化型	非完全 カプセル化型
	重要度 低	非完全 カプセル化型	非完全 カプセル化型	非完全 カプセル化型
使用可能な オブジェクト		なし	完全カプセル化 オブジェクト	完全カプセル化 オブジェクト、 非完全カプセル 化オブジェクト

【図１５】


オリジナル
Java拡張
プロファイル
携帯電話機用
プロファイル
(MIDP)
液晶付
電話機用
プロ
ファイル
TV用
プロ
ファイル
カー
ナビゲーション
用
プロ
ファイル
CLDC
CDC
J2ME
KVM
JVM
OS
J2SE
J2EE
HotSpot

【図１６】



【書類名】　　要約書

【要約】

【課題】　通信装置に記憶されているデータのセキュリティを確保しつつ、当該データを使用する様々なプログラムを提供できるようにする技術を提供する。

【解決手段】　携帯電話機は、プログラムと該プログラムのトラステッドアプリケーション識別情報を受信し、前記プログラムを実行した場合に使用されるデータを特定し、前記トラステッドアプリケーション識別情報に基づいて、生成するオブジェクトの型（非完全カプセル化型、または完全カプセル化型）を選択して選択された型のオブジェクトを生成し、前記プログラムを実行するときには、前記生成されたオブジェクトのみを使用して前記データを使用する。

【選択図】　　図７

特願２００２－３１６６３５

出　願　人　履　歴　情　報

識別番号　　［３９２０２６６９３］

１．変更年月日　　２０００年　５月１９日
［変更理由］　　名称変更
住所変更
住　所　　東京都千代田区永田町二丁目１１番１号
氏　名　　株式会社エヌ・ティ・ティ・ドコモ